Requirements and Analysis document for the DrinkIT Android application

Kajsa Bjäräng, Viktoria Enderstein, Elin Eriksson, Lisa Fahlbeck, Alice Olsson

24/10-2018
version 4

1 Introduction

Background explaining why this application is needed (besides mandatory in course). What's the problem addressed (use imagination)? What will it do? Who will benefit/use from this application? In what situation will the application be used? Define the application. General characteristics of application.

People are always searching for the next great game to play during parties and predrinks. There are many subpar options available, and the goal is to combine and enhance many of these existing games to make the ultimate party-game.

The game will be customizable when it comes to both players and content. Users will be able to enter the number of players and their names. Users can then choose which categories they want to play with for this round and set the length of the game. The categories are made up of a combination of several different games, all popular in different party settings, and a couple unusual ones to give the game depth and keep it exciting.

During the game players will be randomly chosen to perform different challenges, with the options to either "pass" or "fail", decided by the group. The challenges will be randomized from the chosen categories and will have differing levels of difficulty, reflected in the amount of points they give if the players succeed.

At the end of the game a scoreboard will be shown with the points all players have collected, along with the option to play again, which restarts the game, giving the players the choice to either play immediately again or change chosen settings beforehand.

1.1 Definitions, acronyms, and abbreviations

DrinkIT - The name of the application as well as the name of the main class in the model package.
Challenge - A task to perform for each player when it is their turn, for example answer a question or perform a song or charade.
Category - All challenges are divided into one of nine different categories.

GameRound - GameRound contains all information needed to distinguish and save statistics about any round during the game. It contains a player, a challenge, whether the challenge was suceeded or failed and a static list of gameround of the playedrounds.
Gradle - An open-source build automation tool based on Groovy and Kootlin.
Travis - Automatically builds the code and sees if everything still works after a commit to GitHub. Otherwise it report that something went wrong, which is good when several people work with the same project.
Android Studio - An integrated development environment (IDE). It's the official IDE for Google's Android Operating system.

2 Requirements

2.1 User Stories

In order to get a working game these five most important user stories were implemented.

2.1.1 As a user, I want to add a player, so I can have multiple players.

Acceptance Criteria:

- The player exists in the list of players
- It is not possible to add the same player twice

Tasks:

- There is a form to enter a name
- There is a button to add the player

2.1.2 As a user, I want to be able to choose categories, so I can customise the game with challenges I like.

Acceptance Criteria:

- It is possible to see all different categories
- It is possible to only choose one category
- It is possible to choose multiple categories
- It is possible to choose any number of categories
- Only challenges from the chosen categories will be shown during the game.
- It's possible to remove a chosen category

Tasks:

- There exists a number of categories
- If a category is chosen there is a visual feedback
- There are different buttons to add different categories to the game
- There is a button to chose all categories immediately

2.1.3 As a user I would like to choose an approximate duration of the game, so I can decide for how long I would like to play.

Acceptance Criteria:

- When a game is started an option to choose the duration of the game is displayed.
- There are three options to choose from
- Three of the options are set amounts of rounds (short, middle, long game)
- Depending of the number of players and the chosen option for duration an even number of rounds is calculated.
- The calculated number of rounds, depending of the number of players and the chosen option is the exact number of rounds played before the game is finished.

Tasks:

- There are 3 buttons to choose the duration of the game
- The number of rounds is set according to the chosen duration button and the amount of players.

2.1.4 As a user, I want to see a challenge when I play so I can do the challenge.

Acceptance Criteria:

- I can see a card on screen each round
- A player is chosen to play each specific challenge
- The difficulty of the challenge is shown on screen
- The challenge shown belongs to one of the chosen categories
- The challenge shown has not been shown before during the active game

Tasks:

- Randomize a player
- Show the name of the player on the display.
- Randomize a task.
- Show the task on the display

2.1.5 As a user, I want different challenges to appear each round, so I can be sure I don't have to do the same challenge again during the same game.

Acceptance Criteria:

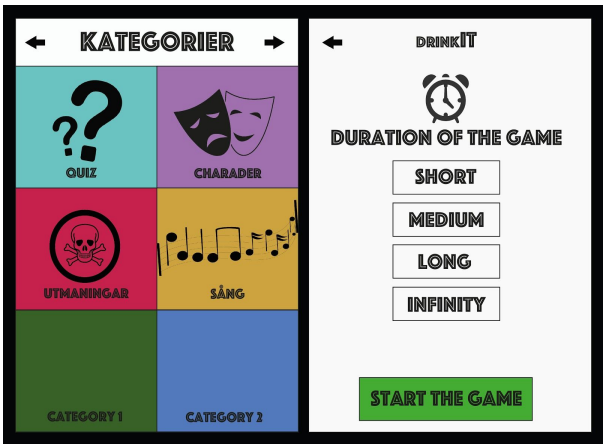
- A challenge shows up for each round
- All challenges that are from the chosen categories are available to appear in the beginning of the game
- Only challenges that are from the chosen categories and that have not been played are available to appear in the end of the game
- As long as there are unplayed challenges available in a chosen category, the players only see new challenges instead of the same challenge twice
- A player will never play the same challenge twice

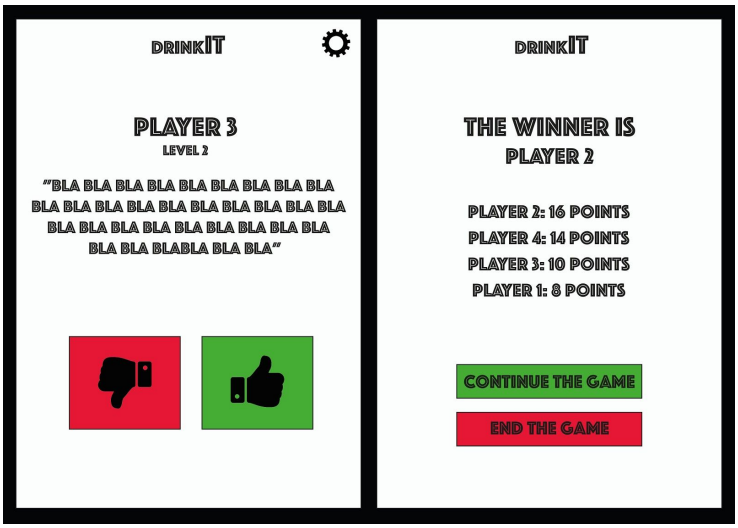
Tasks:

- When a challenge has been played, it is removed from possible challenges to appear in coming rounds, until the list of challenges in a category run out
- Statistics are stored showing which player has played which rounds, so make sure that even if a challenge is shown twice, it doesn't happen to the same player

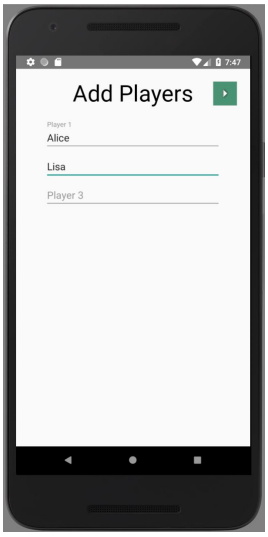
2.2 User interface

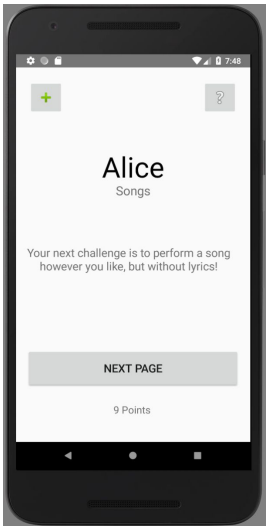
Sketches before code implementation:

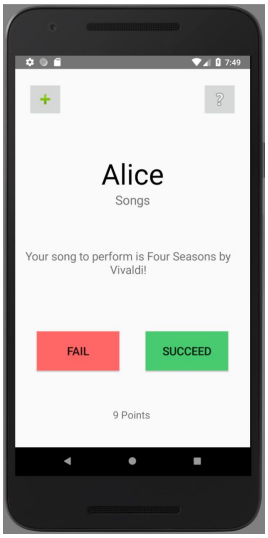


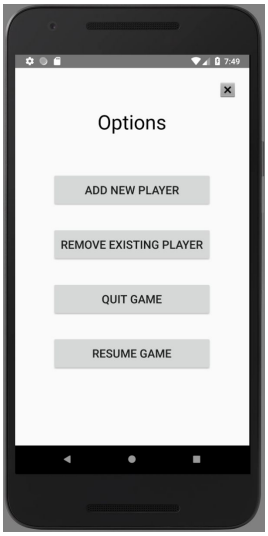


Implemented graphical user interface:

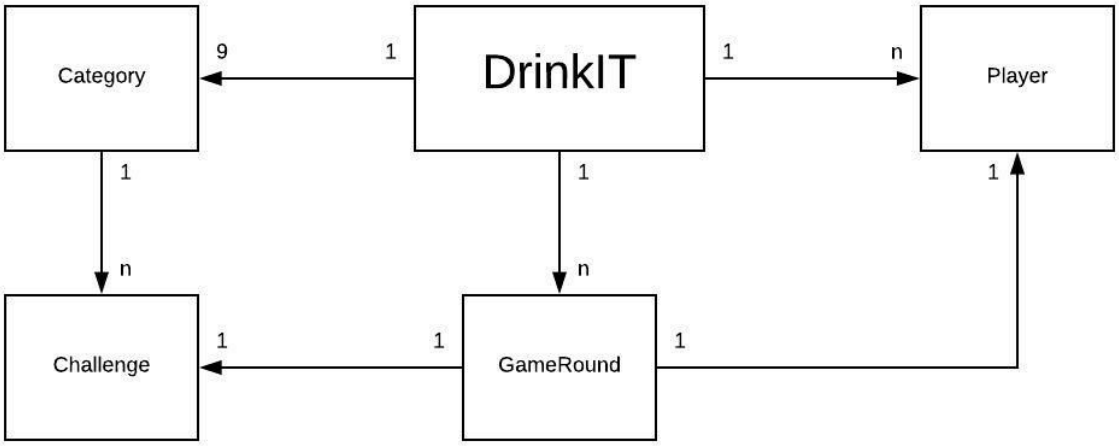








3 Domain model



3.1 Class responsibilities
The application consists of a single component. The model package of the application is represented in the domain model. DrinkIT is the main class in the model and handles most of the functionality for this package and also the communication from the model package and out to the controller.

DrinkIT contains of a list of players which each player has a name and a point. The class also contains a list of 9 categories which can be set to active and inactive depending on the players choice. Each category contains a name, instruction, the state of the category and a list of challenges.

The class GameRound connects the challenges and the players to make a game round. GameRound randomly choose one player and one challenge from an active category and present that to the player by DrinkIT that is connect to the controller which in turn is connected to the view and the information reaches the player. GameRound also has a static list of gamerouns which keeps the playedrounds to be able to keep statistics and for the game to control that one player doesn't get the same challenge twice.

4 References

none.